

AMENDMENTS TO THE CLAIMS**WHAT IS CLAIMED IS:**

1. (Currently Amended) A In a computing device, a method comprising:
providing a definition of a function associated with a first language; and
creating description information about the function from the definition of a function
associated with a first language ~~processing a definition of a function associated with a first~~
~~language to create description information about the function, wherein the description~~
~~information being sufficient to enable enables~~ translation of a call to the function in the first
language into a call to a corresponding function in a second language without requiring
processing of the definition of the function.
2. (Original) The method of claim 1, further comprising: storing the description information in a file of description items.
3. (Currently Amended) The method of claim 1, wherein ~~processing the definition of the~~
~~function~~ creating description information about the function comprises: examining the definition of the function associated with the first language; and deriving information about the function; and using the derived information to translate the call to the function into a call to a
corresponding function in the second language.
4. (Currently Amended) The method of claim 31, further comprising: using the derived
information to create the description information translating a call to a function in the first
language into a call to a function in the second language, without processing of the definition of
the function, using the description information.
5. (Currently Amended) The method of claim 34, further comprising: storing the a translated function in the second language in a library of entries.
6. (Currently Amended) The method of claim 1, in which ~~processing the definition of the~~
~~function~~ creating description information about the function comprises: deriving a number of

declared formal inputs to the function.

7. (Currently Amended) The method of claim 1, in which ~~processing the definition of the function creating description information about the function~~ comprises: deriving a number of declared formal outputs to the function.

8. (Currently Amended) The method of claim 1, in which ~~processing the definition of the function creating description information about the function~~ comprises: deriving a scope of the function.

9. (Currently Amended) The method of claim 1, in which ~~processing the definition of the function creating description information about the function~~ comprises: determining whether the function accepts a variable number of arguments.

10. (Currently Amended) The method of claim 1, in which ~~processing the definition of the function creating description information about the function~~ comprises: determining whether the function returns a variable number of results.

11. (Currently Amended) A In a computing device, a method comprising: providing a file of description items, each item including description information about a function associated with a first language, wherein the description information being sufficient to enable enables translation of a call to the function in a first language into a call to a corresponding function in a second language without requiring processing of the definition of the function; and using the file of description items to translate a first program file into a second program file.

12. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a declared number of formal inputs to the function.

13. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a declared number of formal outputs to the function.

14. (Original) The method of claim 11, wherein the description information about the function

comprises: a descriptor identifying a scope of the function.

15. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying an acceptance of a variable input argument list into the function.

16. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a return of a variable output argument list from the function.

17. (Currently Amended) The method of claim 11, wherein using the file of description items comprises: for each call to a function in the first program file, retrieving an item from the file of description items; using the description information in the item to translate the call to the function in the first language into a call to a corresponding function in the second language; and storing the translated function call in the second program file.

18. (Original) The method of claim 11, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

19. (Original) The method of claim 11, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

20. (Original) The method of claim 11, wherein using the file of description items comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.

21. (Currently Amended) ~~A~~ In a computing device, a method comprising: providing a library file including functions defined by a first language; processing the library file to create creating a function library and a description file from the library file, the function library including one or

more functions defined by a second language, each function in the function library being a translated version of a function in the library file, and the description file including description information about each function in the library file, wherein the description information being sufficient to enable enables translation of a call to the function in the first language into a call to a corresponding function in the second language without requiring processing of the definition of the function; and using the description file to translate a program file from the first language into the second language, wherein each call in the program file to a function in the library file is translated into a call to a corresponding function in the second language.

22. (Currently Amended) The method of claim 21, wherein processing the library file creating a function library comprises: translating the call to each function in the library file into a call to a corresponding function in the second language; and creating a function library including the a translated version of each function in the library file.

23. (Currently Amended) The method of claim 22, further comprising wherein creating a creating description file comprises: examining the definition of each function in the library file; and deriving information about each function; and using the derived information to translate the call to each function into a call to a corresponding function in the second language.

24. (Currently Amended) The method of claim 23, further comprising: using the derived information about each function to create the description information; and creating a description file including description information about each function in the library file.

25. (Original) The method of claim 21, wherein using the description file comprises: for each call in the program file to a function in the library file, retrieving the description information about the function from the description file; and using the description information to translate the call to the function in the first language into a call to a corresponding function in the second language.

26. (Original) The method of claim 21, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the

function.

27. (Original) The method of claim 21, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

28. (Original) The method of claim 21, wherein using the description file comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.

29. (Currently Amended) A computer program product, tangibly stored on a computer-readable medium, for creating a data file, the product comprising instructions operable to cause a programmable processor to: obtain a definition of a function associated with a first language; and process a definition of a function associated with a first language to create description information about the function from the definition of the function associated with a first language, wherein the description information being sufficient to enable enables translation of a call to the function in the first language into a call to a corresponding function in a second language without requiring processing of the definition of the function.

30. (Original) The product of claim 29, further comprising instructions operable to cause a programmable processor to: store the description information in a file of description items.

31. (Currently Amended) The product of claim 29, wherein processing the definition of the function creating description information comprises: examining the definition of the function associated with the first language; and deriving information about the function; and using the derived information to translate the call to the function into a call to a corresponding function in the second language.

32. (Original) The product of claim 31, further comprising instructions operable to cause a programmable processor to: use the derived information to create the description information.

33. (Currently Amended) The product of claim 31, further comprising instructions operable to

cause a programmable processor to: ~~store the translated function in the second language in a library of entries~~ translate a call to a function in the first language into a call to a function in the second language, without processing of the definition of the function, using the description information.

34. (Currently Amended) The product of claim 29, in which ~~processing the definition of the function creating description information~~ comprises: deriving a number of declared formal inputs to the function.

35. (Currently Amended) The product of claim 29, in which ~~processing the definition of the function creating description information~~ comprises: deriving a number of declared formal outputs to the function.

36. (Currently Amended) The product of claim 29, in which ~~processing the definition of the function creating description information~~ comprises: deriving a scope of the function.

37. (Currently Amended) The product of claim 29, in which ~~processing the definition of the function creating description information~~ comprises: determining whether the function accepts a variable number of arguments.

38. (Currently Amended) The product of claim 29, in which ~~processing the definition of the function creating description information~~ comprises: determining whether the function returns a variable number of results.

39. (Currently Amended) A product, stored on a machine-readable medium, for translating a program file, the product comprising instructions operable to cause a processor to: provide a file of description items, each item including description information about a function associated with a first language, the description information ~~being sufficient to enable enabling~~ translation of a call to the function into a call to a corresponding function in a second language without requiring processing of the definition of the function; and use the file of description items to translate a first program file into a second program file.

40. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a declared number of formal inputs to the function.

41. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a declared number of formal outputs to the function.

42. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a scope of the function.

43. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying an acceptance of a variable input argument list into the function.

44. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a return of a variable output argument list from the function.

45. (Currently Amended) The product of claim 39, wherein using the file of description items comprises: for each call to a function in the first program file, retrieving an item from the file of description items; using the description information in the item to translate the call to the function in the first language into a call to a corresponding function in the second language; and storing the translated function call in the second program file.

46. (Original) The product of claim 39, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

47. (Original) The product of claim 39, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

48. (Original) The product of claim 39, wherein using the file of description items comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.

49. (Currently Amended) A computer program product, tangibly stored on a computer-readable medium, for translating function calls, the product comprising instructions operable to cause a programmable processor to: provide a library file including functions defined by a first language; ~~process the library file to create a function library and a description file from the library file~~, the function library including one or more functions defined by a second language, each function in the function library being a translated version of a function in the library file, and the description file including description information about each function in the library file, ~~wherein the description information being sufficient to enable enables~~ translation of a call to the function ~~in the first language~~ into a call to a corresponding function in the second language without requiring processing of the definition of the function; and use the description file to translate a program file from the first language into the second language, wherein each call in the program file to a function in the library file is translated into a call to a corresponding function in the second language.

50. (Currently Amended) The product of claim 49, wherein ~~processing the library file creating a function library~~ comprises: translating the call to each function in the library file into a call to a corresponding function in the second language; ~~and creating a function library including the a translated version of each function in the library file~~.

51. (Original) The product of claim 49, ~~further comprising wherein creating a description file comprises~~: examining the definition of each function in the library file; ~~and deriving information about each function; and using the derived information to translate the call to each function into a call to a corresponding function in the second language~~.

52. (Original) The product of claim 51, further comprising: using the derived information about each function to create the description information; and creating a description file including description information about each function in the library file.

53. (Original) The product of claim 49, wherein using the description file comprises: for each call in the program file to a function in the library file, retrieving the description information about the function from the description file; and using the description information to translate the call to the function in the first language into a call to a corresponding function in the second language.

54. (Original) The product of claim 49, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

55. (Original) The product of claim 49, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

56. (Original) The product of claim 49, wherein using the description file comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.